In this lab session, you will learn about:

- 2D FDTD

- Stability Conditions

- Boundary Conditions

As discussed in the class, we are only going to compute only one field component at a time. Moreover, the corner points are not considered when it comes to computing the fields. This is because of the *wedge condition* due to which the fields shoot to infinity. The last observation is that we need to space the computation parts uniformly across the whole region of space.

In this lab session, we look at the higher dimension cases of FDTD a bit closely. We will be using some **MATLAB** routines for this purpose. One marked difference is that we are going to define the update coefficients so that material properties can be introduced more efficiently.

Using the discretisation approach that we learnt in the class, we can represent any point in space that is transformed from a continuous domain to a discrete domain $(i\Delta x, j\Delta y, k\Delta z)$.

After doing this, we can then express any function u in the finite difference notation in terms of space and time as:

$$u(i\Delta x, j\Delta y, k\Delta z, n\Delta t) = u_{i,j,k}^n \qquad (1)$$

As shown in Fig. 1, we can programatically express the space coordinates x,y, and z with the variable i, j, and k respectively. For time t , the variable n is reserved.
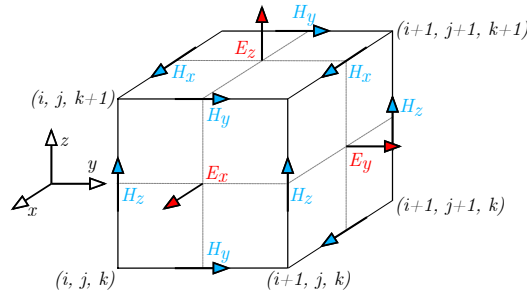

Figure 1: The Yee Lattice showing the space indices and field computation points.

The spacing chosen by Yee in his algorithm produces a 1/2 increment calculation in space:

$$\frac{\partial u(i\Delta x, j\Delta y, k\Delta z, n\Delta y)}{\partial x} = \frac{u_{i+1/2,j,k}^n - u_{i-1/2,j,k}^n}{\Delta x} + \mathcal{O}\left[(\Delta x)^2\right] \qquad (2)$$

Similarly, the time derivative was also chosen to be at 1/2 time increments:

$$\frac{\partial u(i\Delta x, j\Delta y, k\Delta z, n\Delta y)}{\partial t} = \frac{u_{i,j,k}^{n+1/2} - u_{i,j,k}^{n-1/2}}{\Delta t} + \mathcal{O}\left[(\Delta t)^2\right] \qquad (3)$$

1 Finite Difference Time-Domain method

For the sake of complying with duality we introduced a magnetic current term in the Maxwell's equations. Although, there is no physical basis for it, yet the term is introduced so that dual of the electric current is present in the Ampere's law:

$$\frac{\partial \vec{\mathbf{E}}}{\partial t} = \frac{1}{\varepsilon} \boldsymbol{\nabla} \times \vec{\mathbf{H}} - \frac{\sigma}{\varepsilon} \vec{\mathbf{E}} \qquad (4a)$$

$$\frac{\partial \vec{\mathbf{H}}}{\partial t} = -\frac{1}{\mu} \boldsymbol{\nabla} \times \vec{\mathbf{E}} - -\frac{\rho'}{\mu} \vec{\mathbf{H}} \qquad (4b)$$

where the terms ρ', and σ refer to the magnetic resistivity and electrical conductivity. In three dimensions, the magnetic fields from (4) can be written as:

$$\frac{\partial H_x}{\partial t} = \frac{1}{\mu}\left(\frac{\partial E_y}{\partial z} - \frac{\partial E_z}{\partial y} - \rho' H_x\right) \qquad (5a)$$

$$\frac{\partial H_y}{\partial t} = \frac{1}{\mu}\left(\frac{\partial E_z}{\partial x} - \frac{\partial E_x}{\partial z} - \rho' H_y\right) \qquad (5b)$$

$$\frac{\partial H_z}{\partial t} = \frac{1}{\mu}\left(\frac{\partial E_x}{\partial y} - \frac{\partial E_y}{\partial x} - \rho' H_z\right) \qquad (5c)$$

and the electric fields as:

$$\frac{\partial E_x}{\partial t} = \frac{1}{\varepsilon}\left(\frac{\partial H_z}{\partial y} - \frac{\partial H_y}{\partial z} - \sigma E_x\right) \tag{6a}$$

$$\frac{\partial E_y}{\partial t} = \frac{1}{\varepsilon}\left(\frac{\partial H_x}{\partial z} - \frac{\partial H_z}{\partial x} - \sigma E_y\right) \tag{6b}$$

$$\frac{\partial E_z}{\partial t} = \frac{1}{\varepsilon}\left(\frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y} - \sigma E_z\right) \tag{6c}$$

As an example, the central difference approximation for (5a) becomes:

$$\frac{H_x|_{i,j,k}^{n+1/2} - H_x|_{i,j,k}^{n-1/2}}{\Delta t} = \frac{1}{\mu_{i,j,k}}\left[\frac{E_y|_{i,j,k+1/2}^{n} - E_y|_{i,j,k-1/2}^{n}}{\Delta z}\right.$$
$$\left. - \frac{E_z|_{i,j+1/2,k}^{n} - E_y|_{i,j-1/2,k}^{n}}{\Delta y} - \rho'_{i,j,k}\, H_x|_{i,j,k}^{n}\right] \tag{7}$$

Notice that in (7), $\vec{\mathbf{E}}$ and $\vec{\mathbf{H}}$ exist $1/2$ time step apart. As the electric fields occupy the time step n, the magnetic fields only exist at half-time step $n-1/2$, or $n+1/2$. This means that $H_x|_{i,j,k}^{n}$ needs to be *approximated*. Often times, we just use the average:

$$H_x|_{i,j,k}^{n} = \frac{H_x|_{i,j,k}^{n+1/2} + H_x|_{i,j,k}^{n-1/2}}{2} \tag{8}$$

From (7) and (8) we get:

$$H_x|_{i,j,k}^{n+1/2} - H_x|_{i,j,k}^{n-1/2} = \frac{\Delta t}{\mu_{i,j,k}}\left[\frac{E_y|_{i,j,k+1/2}^{n} - E_y|_{i,j,k-1/2}^{n}}{\Delta z} - \right.$$
$$\left. \frac{E_z|_{i,j+1/2,k}^{n} - E_y|_{i,j-1/2,k}^{n}}{\Delta y}\right] \tag{9}$$

After collecting $H_x|_{i,j,k}^{n+1/2}$ and $H_x|_{i,j,k}^{n-1/2}$, we get:

$$\left(1 + \frac{\Delta t}{\mu_{i,j,k}} \frac{\rho'_{i,j,k}}{2}\right) H_x|_{i,j,k}^{n+1/2} = \left(1 - \frac{\Delta t}{\mu_{i,j,k}} \frac{\rho'_{i,j,k}}{2}\right) H_x|_{i,j,k}^{n-1/2}$$

$$+ \frac{\Delta t}{\mu_{i,j,k}} \left[\frac{E_y|_{i,j,k+1/2}^{n} - E_y|_{i,j,k-1/2}^{n}}{\Delta z} - \frac{E_z|_{i,j+1/2,k}^{n} - E_y|_{i,j-1/2,k}^{n}}{\Delta y}\right] \quad (10)$$

Dividing (**??**) by dividing by $\left(1 + \frac{\Delta t}{\mu_{i,j,k}} \frac{\rho'_{i,j,k}}{2}\right)$, we get:

$$H_x|_{i,j,k}^{n+1/2} = \underbrace{\left(\frac{1 - \frac{\rho'_{i,j,k}\Delta t}{2\mu_{i,j,k}}}{1 + \frac{\rho'_{i,j,k}\Delta t}{2\mu_{i,j,k}}}\right)}_{D_{a,\,H_x|_{i,j,k}}} H_x|_{i,j,k}^{n-1/2}$$

$$+ \underbrace{\left(\frac{\frac{\Delta t}{\mu_{i,j,k}}}{1 + \frac{\rho'_{i,j,k}\Delta t}{2\mu_{i,j,k}}}\right)}_{D_{b,\,H_x|_{i,j,k}}} \left[\frac{E_y|_{i,j,k+1/2}^{n} - E_y|_{i,j,k-1/2}^{n}}{\Delta z} - \right.$$

$$\left. \frac{E_z|_{i,j+1/2,k}^{n} - E_y|_{i,j-1/2,k}^{n}}{\Delta y} - \rho'_{i,j,k}\frac{H_x|_{i,j,k}^{n+1/2} + H_x|_{i,j,k}^{n-1/2}}{2}\right] \quad (11)$$

Using the same procedures, we can find out the expressions for other components of the magnetic field:

$$H_y|_{i,j,k}^{n+1/2} = \underbrace{\left(\frac{1 - \frac{\rho'_{i,j,k}\Delta t}{2\mu_{i,j,k}}}{1 + \frac{\rho'_{i,j,k}\Delta t}{2\mu_{i,j,k}}} \right)}_{D_{a,\,H_y}|_{i,j,k}} H_y|_{i,j,k}^{n-1/2}$$

$$+ \underbrace{\left(\frac{\frac{\Delta t}{\mu_{i,j,k}}}{1 + \frac{\rho'_{i,j,k}\Delta t}{2\mu_{i,j,k}}} \right)}_{D_{b,\,H_y}|_{i,j,k}} \left[\frac{E_z|_{i+1/2,j,k}^{n} - E_z|_{i-1/2,j,k}^{n}}{\Delta x} \right.$$

$$\left. - \frac{E_x|_{i,j,k+1/2}^{n} - E_x|_{i,j,k-1/2}^{n}}{\Delta z} \right] \qquad (12a)$$

$$H_z|_{i,j,k}^{n+1/2} \; c \; \underbrace{\left(\frac{1 - \frac{\rho'_{i,j,k}\Delta t}{2\mu_{i,j,k}}}{1 + \frac{\rho'_{i,j,k}\Delta t}{2\mu_{i,j,k}}} \right)}_{D_{a,\,H_z}|_{i,j,k}} H_z|_{i,j,k}^{n-1/2} +$$

$$\underbrace{\left(\frac{\frac{\Delta t}{\mu_{i,j,k}}}{1 + \frac{\rho'_{i,j,k}\Delta t}{2\mu_{i,j,k}}} \right)}_{D_{b,\,H_z}|_{i,j,k}} \left[\frac{E_x|_{i,j+1/2,k}^{n} - E_x|_{i,j-1/2,k}^{n}}{\Delta y} \right.$$

$$\left. - \frac{E_y|_{i+1/2,j,k}^{n} - E_y|_{i-1/2,j,k}^{n}}{\Delta x} \right] \qquad (12b)$$

Similarly, the electric field components are:

$$E_x|_{i,j,k}^{n+1} = \underbrace{\left(\frac{1 - \frac{\sigma_{i,j,k}\Delta t}{2\varepsilon_{i,j,k}}}{1 + \frac{\sigma_{i,j,k}\Delta t}{2\varepsilon_{i,j,k}}}\right)}_{C_{a,\,E_x}|_{i,j,k}} E_x|_{i,j,k}^{n}$$

$$+ \underbrace{\left(\frac{\frac{\Delta t}{\varepsilon_{i,j,k}}}{1 + \frac{\sigma_{i,j,k}\Delta t}{2\varepsilon_{i,j,k}}}\right)}_{C_{b,\,E_x}|_{i,j,k}} \left[\frac{H_z|_{i,j+1/2,k}^{n+1/2} - H_z|_{i,j-1/2,k}^{n+1/2}}{\Delta y}\right.$$

$$\left. - \frac{H_y|_{i,j,k+1/2}^{n+1/2} - H_y|_{i,j,k-1/2}^{n+1/2}}{\Delta z}\right] \tag{13a}$$

$$E_y|_{i,j,k}^{n+1} = \underbrace{\left(\frac{1 - \frac{\sigma_{i,j,k}\Delta t}{2\varepsilon_{i,j,k}}}{1 + \frac{\sigma_{i,j,k}\Delta t}{2\varepsilon_{i,j,k}}}\right)}_{C_{a,\,E_y}|_{i,j,k}} E_y|_{i,j,k}^{n}$$

$$+ \underbrace{\left(\frac{\frac{\Delta t}{\varepsilon_{i,j,k}}}{1 + \frac{\sigma_{i,j,k}\Delta t}{2\varepsilon_{i,j,k}}}\right)}_{C_{b,\,E_y}|_{i,j,k}} \left[\frac{H_x|_{i,j,k+1/2}^{n+1/2} - H_x|_{i,j,k-1/2}^{n+1/2}}{\Delta z}\right.$$

$$\left. - \frac{H_z|_{i+1/2,j,k}^{n+1/2} - H_z|_{i-1/2,j,k}^{n+1/2}}{\Delta x}\right] \tag{13b}$$

$$E_z|_{i,j,k}^{n+1} = \underbrace{\left(\frac{1 - \frac{\sigma_{i,j,k}\Delta t}{2\varepsilon_{i,j,k}}}{1 + \frac{\sigma_{i,j,k}\Delta t}{2\varepsilon_{i,j,k}}}\right)}_{C_{a,\,E_z}|_{i,j,k}} E_z|_{i,j,k}^{n}$$

$$+ \underbrace{\left(\frac{\frac{\Delta t}{\varepsilon_{i,j,k}}}{1 + \frac{\sigma_{i,j,k}\Delta t}{2\varepsilon_{i,j,k}}}\right)}_{C_{b,\,E_z}|_{i,j,k}} \left[\frac{H_y|_{i+1/2,j,k}^{n+1/2} - H_y|_{i-1/2,j,k}^{n+1/2}}{\Delta x}\right.$$

$$\left. - \frac{H_x|_{i,j+1/2,k}^{n+1/2} - H_x|_{i,j-1/2,k}^{n+1/2}}{\Delta y}\right] \tag{13c}$$

Our goal will be to define the update coefficients C and D when we need to create inhomogeneous medium simulations. As an example, when the conductivity σ and magnetic resistivity ρ' are both equal to zero, the coefficients become,

$$D_{a,\,H_{x,y,z}}|_{i,j,k} = 1 \tag{14a}$$

$$D_{b,\,H_{x,y,z}}|_{i,j,k} = \frac{\Delta t}{\mu_{i,j,k}} \tag{14b}$$

$$C_{a,\,E_{x,y,z}}|_{i,j,k} = 1 \tag{14c}$$

$$C_{b,\,E_{x,y,z}}|_{i,j,k} = \frac{\Delta t}{\varepsilon_{i,j,k}} \tag{14d}$$

$$\tag{14e}$$

For the $2D$ case we assume ($\frac{\partial}{\partial z} = 0$), and then write the uncoupled expressions as TM and TE modes:

TM mode:

$$\frac{\partial H_x}{\partial t} = \frac{1}{\mu}\left(-\frac{\partial E_z}{\partial y} - \rho' H_x\right) \tag{15a}$$

$$\frac{\partial H_y}{\partial t} = \frac{1}{\mu}\left(\frac{\partial E_z}{\partial x} - \rho' H_y\right) \tag{15b}$$

$$\frac{\partial E_z}{\partial t} = \frac{1}{\varepsilon}\left(\frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y} - \sigma E_z\right) \tag{15c}$$

TE mode:

$$\frac{\partial E_x}{\partial t} = \frac{1}{\varepsilon}\left(\frac{\partial H_z}{\partial y} - \sigma E_x\right) \tag{16a}$$

$$\frac{\partial E_y}{\partial t} = \frac{1}{\varepsilon}\left(-\frac{\partial H_z}{\partial x} - \sigma E_y\right) \tag{16b}$$

$$\frac{\partial H_z}{\partial t} = \frac{1}{\mu}\left(\frac{\partial E_x}{\partial y} - \frac{\partial E_y}{\partial x} - \rho' H_z\right) \tag{16c}$$

Using the coefficients in (14), the 2D TM mode central difference equations from (15) are:

$$H_x|_{i,j}^{n+1/2} = H_x|_{i,j}^{n-1/2} - \frac{\Delta t}{\mu_{i,j}} \left[\frac{E_z|_{i,j+1/2}^{n} - E_z|_{i,j-1/k2}^{n}}{\Delta y} \right] \tag{17a}$$

$$H_y|_{i,j}^{n+1/2} = H_y|_{i,j}^{n-1/2} - \frac{\Delta t}{\mu_{i,j}} \left[\frac{E_z|_{i+1/2,j}^{n} - E_z|_{i-1/2,j}^{n}}{\Delta x} \right] \tag{17b}$$

$$E_z|_{i,j}^{n+1} = E_z|_{i,j}^{n} - \frac{\Delta t}{\varepsilon_{i,j}} \left[\frac{H_y|_{i+1/2,j}^{n+1/2} - H_y|_{i-1/2,j}^{n+1/2}}{\Delta x} - \frac{H_x|_{i,j+1/2}^{n+1/2} - H_x|_{i,j-1/2}^{n+1/2}}{\Delta y} \right] \tag{17c}$$

2 Programming the equations

We look to solve the 2D TM mode case; the rest of the cases follow a similar approach.

1. Before we perform the time-stepping of the fields, we compute all the update coefficients $C_{a,b}$ and $D_{a,b}$ and store them in an array. These will remain unchanged.

2. We compute the magnetic fields 2 time-steps back, ie. $H_x|_{i,j}^{n-1/2}$ and $H_y|_{i,j}^{n-1/2}$ and store them in the array Hx(i,j) and Hy(i,j) respectively.

3. We then compute the electric field Ez(i,j) 1 time-step back ($E_z|_{i,j}^{n}$).

4. At the current time, which is always $n+1/2$, the magnetic field components, $H_x|_{i,j}^{n+1/2}$ and $H_y|_{i,j}^{n+1/2}$ are computed and the magnetic field array is updated accordingly.

5. In the next time step, $E_z|_{i,j}^{n+1}$ is computed and stored in the Ez(i,j) array where we replace the value $E_z|_{i,j}^{n}$ with the new one.

6. As the updating of the E and H actually implements the time-stepping, we need to have a time index explicitly.

Now let's look at a **Matlab** routine that implements the 2D TM mode fields in a rectangular space with perfect electric (PEC) boundaries.

```matlab
% This program computes and plots the TM mode fields in
% 2D rectagnular space with PEC boundaries using
% Yee's FDTD Algorithm.
%
%% Define Parameters
c = 2.99792458e8; % Speed of light
eps0 = 8.854187817e-12; % Free space permittivity
xmu0 = 4*pi*1e-7; % Free space permeability
%
%%  Define Space Region
a = 10; b = 10; % Space Dimensions
% Number of grid points in x and y direction
nx = 100; ny = 100;
% Convert to integer in-case of odd number
nx2 = ceil(nx/2); ny2 = ceil(ny/2);
nt = 200; % Number of time-steps
nskip = 5; % Time-steps to skip while recording
nsnap = ceil(nt/nskip);
%
%% Create a Gaussian Excitation Source
xndec = 10.0; % Mean
xn0 = 4*xndec; % Variance
%
%% Initialize
%
Ez = zeros( nx, ny ); %  Z E-field initialize to zero
Hx = zeros( nx, ny ); %  X H-field initialize to zero
Hy = zeros( nx, ny ); %  Y H-field initialize to zero

dx = a/(nx - 1); % Define spatial step
dy = dx; %
ds = dx;

% ************************
% ************************
dt = ds/(c*sqrt(2)); % Stability Condition
% ************************
% ************************
```

The approach of the Yee algorithm along with the material definition follows what has been described in [1, pg. 85]. The medium is defined as:

```matlab
%% Medium Definition
%
dte = ones(nx,ny)*dt/(ds*eps0);
dtm = ones(nx,ny)*dt/(ds*xmu0);
Da = ones(nx,ny);
Db = dtm;
Ca = ones(nx,ny);
Cb = dte;
%
```

Next, we implement the Yee algorithm:

```matlab
%% Yee Algorithm
for n = 1:nt
    for i = 1 : nx
        for j = 1 : ny
            if (i == nx2 && j == ny2) % Place source in the center

                Ez(i,j) = exp(-((n-xn0)/(xndec))^2); % Gaussian Source

            elseif (i == 1 || j == 1 || i == nx || j == ny )

                Ez(i,j) = 0; % PEC boundaries at the edges

            else

                Ez(i,j) = Ez(i,j)*Ca(i,j) + Cb(i,j)*(Hy(i,j) - Hy(i-1,j)...
                    - (Hx(i,j) - Hx(i,j-1)));

            end
        end
    end

    for i = 1 : nx
        for j = 1 : ny - 1

            Hx(i,j) = Hx(i,j)*Da(i,j) - Db(i,j)*(Ez(i,j+1) - Ez(i,j));

        end
    end

    for i = 1 : nx - 1
        for j = 1 : ny

            Hy(i,j) = Hy(i,j)*Da(i,j) + Db(i,j)*(Ez(i+1,j) - Ez(i,j));

        end
    end
end
```

We can use the plotting routine below to visualise the electric fields:

```matlab
    %% Plot Routine
%
xd = 0:ds:a;
[xdg, ydg] = meshgrid(xd, xd);
figure(1)
EzSurf = surf(xdg,ydg,Ez,'EdgeColor','interp','FaceLighting','gouraud');
shading interp % Avoid jittered shading
set(gcf,'Color','white'); % Set background color to white
set (gca,'FontName','times new roman') % Set axes fonts to Times New Roman
box on %
colorbar % Show colorbar on the right of the figure
grid on
colormap('jet')
material dull % Set reflectivity of the surface to dull
% Add three lights below tom improve visuals
```

```matlab
h = light;
light('Position',[10 0 1]);
light('Position',[-190 -120 1]);
light('Position',[0 0 1]);
title(['E-field at Time Step ',int2str(n)])
% Create Title with mode numbers
matFileName = sprintf('E_Field_%d', n);

if rem(n,nskip) == 0

    M_PEC(:,n/nskip) = getframe; % Capture frames to create a movie

end

if n == 20 || n == 60 || n == 80 || n == 100 ||...
        n == 120 || n == 140 || n == 160
    saveas(gcf,[matFileName,'.eps'],'epsc') % Save visualizations
end
```

Running the Matlab file, we should get a 2D Gaussian source starting from the origin and expanding outwards as seen in the figure below:

Exercise

Modify the above program to simulate the TE mode. The expressions will be changed accordingly as provided in (16).

3 Stability

There has been extensive work done regarding the stability of the FDTD simulations. Stability ensures that any sources created don't shoot up in value as the simulation is progressed in time. As we saw in the lectures, the Courant number is a criterion for the stability of the FDTD simulation. It is different for 1-, 2-, and 3- dimensions. In order to find the stability conditions, the difference based Maxwell equations are treated as an eigenvalue problem and the eigenmodes determine the numerical stability. Essentially, we want to avoid the uncontrolled growth of all the possible spatial modes in the FDTD grid. As an example, a mode consisting of a sinusoidal function must have values in the limit $-1, \ldots, 1$. Generally speaking, if we assume equal spatial spacing, ie, $\Delta_x = \Delta_y = \Delta_z = \Delta$, then a good rule of thumb for stability condition is:
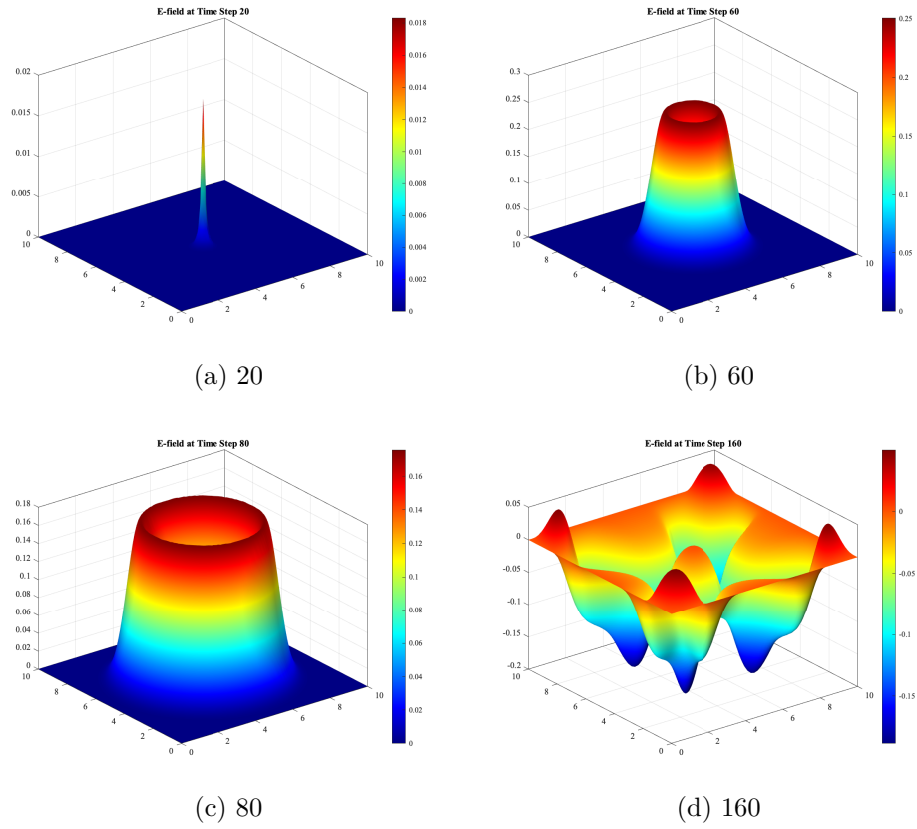
(a) 20

(b) 60

(c) 80

(d) 160

Figure 2: The E_z component visualised at different time intervals.

$$\Delta_t \leq \frac{\Delta}{2c} \tag{18}$$

where c is the speed of light in the medium.

Exercise

Modify the stability condition in the Matlab routine (`Program 1.m`) to the following and plot the results at time instants 20, 40, 80, and 160.

```
% *************************
% *************************
dt = 1.3*ds/(c*sqrt(2)); % Stability Condition
% *************************
% *************************
```

Also briefly explain what is happening in the simulation.

4   Yee paper

Let's try to replicate some of the results presented in Yee's seminal paper on FDTD [2]. The paper simulates a 2D TM scattering problem, where there is a perfectly conducting square is placed at the centre. The square can be produced as below:

```
%% Create Structure in the computational space
% *************************
% *************************
function define_media(iflaga)

global nx  ny mxst mxnd myst mynd;
global mediaEz mediaHx mediaHy; %
if (iflaga == 2)

    for  i = 1:nx
        for j = 1:ny
            if (i >= mxst && i <= mxnd)
                if ( j >= myst && j <= mynd)
                    mediaEz(i,j) = 2;
                end
            end
        end
    end

    for  i = 1:nx
        for j = 1:ny
            if (i >= mxst && i <= mxnd)
```

```
                    if ( j >= myst && j <= mynd-1)
                        mediaHx(i,j) = 2;
                    end
                end
            end
        end

        for  i = 1:nx
            for j = 1:ny
                if (i >= mxst && i <= mxnd-1)
                    if ( j >= myst && j <= mynd)
                        mediaHy(i,j) = 2;
                    end
                end
            end
        end
    end

    end
```

Here the variable, `iflaga` either assumes the value 1 (free space) or 2 (PEC). The complete code can be found in the program `Program 2.m`.

Exercise

Please run the program and produce the results for `n = 5` as shown in Figs 3 - 6 in [2].

5   Boundary Conditions

As we saw in class that we require FDTD to model *open* regions where the spatial domain is unbounded. However, practically it is not possible as it requires an infinite amount of computer memory. Therefore, in order to limit the size of the simulation, we create a finite simulation domain with *absorbing boundary conditions* (ABCs).

A class of ABCs work on *extrapolating* the field values at the boundary. One of the most famous ones is the Liao ABC [3] which works on updating the numerical fields in space and time using a Newton backward-difference polynomial.

The idea is that for any point on the grid bounday x_{MAX}, we need to calculate the updated field $u(x_{MAX}, t + \Delta_t)$ using the field values that we already have in the memory. As an example, the update field expressions for a 1D case can be written as:
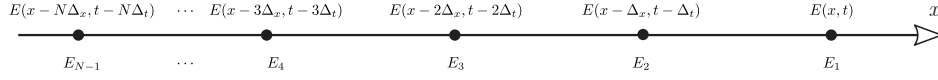
14

Figure 3: The backward difference used to compute the boundary points.

$$\Delta^1 E(x,t) = \Delta^1 E_1 = E_2 - E_1 = E(x,t) - E(x - \Delta_x, t - \Delta_t)$$
$$\Delta^2 E_1 = \Delta^1 E_1 - \Delta^1 E_2$$
$$= E(x,t) - \Delta^1 E(x - \Delta_x, t - \Delta_t)$$
$$= E(x,t) - 2E(x - \Delta_x, t - \Delta_t) + E(x - 2\Delta_x, t - 2\Delta_t) \qquad (19)$$

For the Nth term, we get:

$$\Delta^N E(x,t) = \sum_{j=1}^{N+1} C_{j-1}^N E\left[x - (j-1)\Delta_x, t - (j-1)\Delta_t\right]$$
$$\text{where,}\, C_k^N = \frac{N!}{(N-k)!\,k!} \qquad (20)$$

The *extrapolation* can be understood from Fig. 3 where the point on the right is extrapolated from the neighbouring points, depending on the order of the Liao ABC.

To implement the Liao ABCs in MATLAB, we first define the coefficients C_k^N for a particular order. As an example, the $4 - th$ order coefficients are:

```
    function define_Liao()

global c1 c2 c3 c4 c5 ; % Define material based coefficients
global ABC_order % Order of the Liao ABC
switch ABC_order


    case 4 %% 4th order LIAO ABC Coefficients
        c1 = 4;
        c2 = 6;
        c3 = 4;
        c4 = 1;
        c5 = 0;

    otherwise
```

15

```matlab
        disp('Error: Wrong Value')
    end
end
```

We then implement the Liao ABCs as:

```matlab
%% Implement LIAO ABC
% ************************
% ************************
function Liao_ABC()

global c1 c2 c3 c4 c5 ; % Define material based coefficients
global Ez ; % E and H field components.
global Ez1 Ez2 Ez3 Ez4 Ez5 % For Bubbling of E-fields in Liao ABC
global nx  ny
global ABC_order
% General BC for any order LIAO ABC
    for  j = 1:ny
    Ez(1,j) = c1*Ez1(2,j)-c2*Ez2(3,j)+c3*Ez3(4,j)...
        -c4*Ez4(5,j)+c5*Ez5(6,j);%%%left side
    end
    for j = 1:ny
    Ez(nx,j) = c1*Ez1(nx-1,j)-c2*Ez2(nx-2,j)+c3*Ez3(nx-3,j) ...
        -c4*Ez4(nx-4,j)+c5Ez5(nx-5,j);%%%right side
    end
    for i = 2:nx-1
    Ez(i,1) = c1*Ez1(i,2)-c2*Ez2(i,3)+c3*Ez3(i,4) ...
        -c4*Ez4(i,5)+c5*Ez5(i,6);%%%bottom
    end
    for i = 2:nx-1
    Ez(i,ny) = c1*Ez1(i,ny-1)-c2*Ez2(i,ny-2)+c3*Ez3(i,ny-3) ...
        -c4*Ez4(i,ny-4)+c5*Ez5(i,ny-5);%%%top
    end

    switch ABC_order

        case 5
            Ez5 = Ez4;
            Ez4 = Ez3;
            Ez3 = Ez2;
            Ez2 = Ez1;
            Ez1 = Ez;

        case 4
            Ez4 = Ez3;
            Ez3 = Ez2;
            Ez2 = Ez1;
            Ez1 = Ez;

        case 3
            Ez3=Ez2;
            Ez2=Ez1;
            Ez1=Ez;

        otherwise
            disp('Error: Wrong Value')
```

```
        end
    end
```

In `Program 3.m`, the Liao ABCs are implemented for the same structure as in the Yee paper.

Exercise

Plot the 1D electric fields at `n = 5` and try see how the Figs 3 - 6 in [2] now look like.

Home Exercise

Create a sinusoidal source on the left boundary in a 2D TM case and replace the PEC square in the Yee paper simulation with a dielectric square of $\varepsilon = 4$ of the same size.

6 Code Availability

You can download the codes used in this manual from the Github repository.

References

[1] A. Taflove and S. C. Hagness, *Computational electrodynamics: the finite-difference time-domain method*, 3rd ed., ser. Artech House antennas and propagation library.   Boston: Artech House, 2005.

[2] Kane Yee, "Numerical solution of initial boundary value problems involving maxwell's equations in isotropic media," *IEEE Transactions on Antennas and Propagation*, vol. 14, no. 3, pp. 302–307, May 1966.

[3] Z.-F. Liao, K.-L. Huang, B.-P. Yang, and Y.-F. Yuan, "A TRANSMITTING BOUNDARY FOR TRANSIENT WAVE ANALYSES," *Science in China Series A-Mathematics, Physics, Astronomy & Technological Science*, vol. 27, no. 10, pp. 1063–1076, Oct. 1984. [Online]. Available: /doi/10.1360/ya1984-27-10-1063